Lead Push - External Clients

Housing Lead service has the capability of pushing leads to external services.. Every third party needs to provide a payload,(sample payload is shared below) and the communication works on a particular authentication. Here is the list of supported authentications.

1. CSRF Token based authentication
2. Key based authentication
3. Oath2 token based authentication
4. Static key based authentication
5. No authentication

Details of all the above authentication is given below..

Sample Lead Payload To be Pushed:

Client is expected to provide a payload containing all the lead details needed which will be sent in the Request Body.

A sample payload will look like this:

```
{
  'name': <user's name>,

'email': <user's email>,

'mobile_number': <user's mobile_number>,

'city': <the city lead was submitted in>,

  'project_id' : ,


}
```

Supported Authentications

CSRF Request Handler

This api Request Handler uses HTTP Basic Auth to fetch the CSRF Token needed to call the Lead Push API

- Requires a URL to fetch the token from
- Requires username and password
- The Request timeout here is set to be 10 seconds
- The Client API is expected to return `x-csrf-token` in the Response headers of this API call

Step-1 CSRF Token Request API

- `Request Method: GET`

`API Config Required`

```
{
  'url': <URL to fetch token>,
  'auth': {
    'type': 'Basic',
    'username': <username>,
    'password': <password>,
   }
}
```

Step -2 Lead Push API

Once a CSRF token is fetched from the first request, we prepare the payload and request headers for the Lead to be sent in the Lead Push API

- `Request Method: POST`

Request headers:

```
{
  'headers' => {
    'x-csrf-token': <CSRF Token fetched from the first request>,
    'cache-control': 'no-cache',
    'content-type': 'application/json',

'Cookie': <Cookie data received in Set-Cookie response header in the first request>,

}
}
```

Details:

- The Lead Push API is expected to be protected with Basic Auth and is called with the same credentials as used in the first request
- The Request timeout here is set to be 10 seconds

Key Based Auth Handler

Request Handler makes a POST API call to fetch the token needed to call the Lead Push API

- Requires a URL to fetch the token
- Username, Password and Key (Secret Key) is passed in Request body
- The Client API is expected to return a user token in the Response body of this API call
    - Expected Response Body

      ```
      {

      'UserAuthenticationResult': {
          'data': {
      'UserToken': <token-value>
      }

      }
      }
      ```

- The Request timeout here is set to be 10 seconds

Fetch User Token Request

- `Request Method: POST`

`API Config Required`

```
{
  'url': <URL to fetch token>,
  'Password': <password>,
  'username': <username>,
  'Key': <secret-key>
}
```

Request headers:

```
{
  'headers' => {
    'APPLICATION_ID': <Application ID for the client>,

'content-type': 'application/json',

}
}
```

Lead Push API Config

- `Request Method: POST`

`API Config Required`

```
{
  'url': <URL to push the lead at>
}
```

Once a User token is fetched from the first request, we prepare the payload and request headers for the Lead to be sent in the Lead Push API

Request headers:

```
{
  'headers' => {
    'APPLICATION_ID': <Application ID for the client>,

'content-type': 'application/json',

'USER_TOKEN': <Token received in the first request>

}
}
```

Details:

- The Lead Push API is expected to validate the USER_TOKEN sent and do the needful
- The Request timeout here is set to be 10 seconds

OAuth2 Request Handler

Fetch OAuth Token Request

- `Request Method: POST`

API Config Required

```
{
  'url': <URL to fetch token>,
  'username': <username>,
  'password': <password>,
  'client_id': <client id>

'client_secret': <client secret key>,
  'grant_type': <grant type>

}
```

Request headers:

```
{
  'headers' => {

'content-type': 'application/json'

}
}
```

Details:

Request Handler makes a POST API call to fetch the OAuth token needed to call the Lead Push API

- Requires a URL to fetch the token
- The request body is comprised of these parameters:
  - Username
  - Password
  - Client ID
  - Client Secret
  - Grant Type
- The Client API is expected to return an access token in the Response body of this API call
  - Expected Response Body

    ```
    {

    'access_token': <token-value>

    }
    ```

- The Request timeout here is set to be 10 seconds

Lead Push API Config

- `Request Method: POST`

<u>API Config Required</u>

```
{
  'url': <URL to push the lead at>
}
```

Once a User token is fetched from the first request, we prepare the payload and request headers for the Lead to be sent in the Lead Push API

<u>Request headers:</u>

```
{
  'headers' => {

'content-type': 'application/json',

'Authorization': "Bearer <access token received in the first request>"

}
}
```

<u>Details:</u>

- The Lead Push API is expected to validate the Access Token sent and do the needful
- The Request timeout here is set to be 10 seconds

Static Key Based Auth Handler

In this Key Auth Handler, Static keys are exchanged which are then used for Authenticating every lead Push API request

Lead Push API Config

- `Request Method: POST`

Auth Config Required

```
{
  'vendor': <Vendor Name>,
  'vendory_key': <Vendor Key>

}
```

<u>API Config Required</u>

```
{
  'url': <URL to push the lead at>
}
```

<u>Details:</u>

Request Handler makes a Lead Push POST API call with the request headers and the payload containing the Lead details and the Auth information as specified above

<u>Request headers:</u>

```
{
  'headers' => {

'content-type': 'application/x-www-form-urlencoded'

}

}
```

<u>Details:</u>

- The Lead Push API is expected to validate the Auth keys sent and do the needful

- The Request timeout here is set to be 10 seconds

No Authentication Handler

This is used when no authentication is required from the client side and a POST request with the payload is required to be hit on the URL provided by the client.

Lead Push API Config

- `Request Method: POST`

`API Config Required`

```
{
  'url': <URL to push the lead at>
}
```